Predicting Electrical Energy Output of a Power Plant

Name: Sulaiman Alhussaini
Email: saalhuss@usc.edu
Date:12/3/2021

1. Abstract

In the project, we are going to deal with a real-world dataset that was collected from a combined cycle power plant over 6 years (2006-2011) when the power plant was set to work with full load. This dataset contains 9568 datapoints and four continues features. Our goal is to predict the net hourly electrical energy output (EP) of the plant by finding the best predictive feature and optimizing the parameters of different regression models and compare their performance. A semi-supervised approach will also be conducted including a comparison between the performance of the learnt models using only the labeled data and using the whole data. In addition, noisy data will mostly be present in such industrial problem domains, hence it might be suitable to add a training noise to study the behavior of the models and compare their robustness against noise. The sensitivity to the training size will also be addressed, and how much training data is required for each algorithm in order to still have a reasonable generalization performance.

firstly, a proportion of 20% of the data will be used as a test set to evaluate the performance of the chosen models. The rest of the data will be split into a validation set to find the models' hyperparameter and have an estimate of the test error in the presence of noise or in light training situations, and a training set for model learning. The training and validation sets might be combined to perform cross-validation for model selection instead of validation set approach, and the two approaches will be compared. The outliers and missing data will also be addressed depending on their significance. A final model will be chosen depending mainly on its error performance and generalization, and according to other criteria's such as noise robustness and data hungriness. A co-training semi-supervised approach will be compared to the supervised approach in the presence of limited number of labeled training points, where the semi-supervised approach learns two regressors in two views and they update each other using the most confident predicted points.

2. Introduction

2.1. Problem Type, Statement and Goals

This is a regression problem with the main goal of predicting the net hourly electrical energy output of an electrical power plant. There are some challenges that need to be addressed, including:

(1) Low dimensionality of feature space:
We notice that the data contains only 4 predictors, and they might not be the only factors that naturally affect the output variable, hence we will find out whether we will be able to accurately predict the output variable or not using those limited number of features.

(2) Outliers:
Outliers will generally have negative effect on the statistical and machine learning techniques as they introduce some biases and errors. Therefore, we will identify outliers and evaluate how to treat them.

(3) Nonlinear behaviors:
There might be some non-linear relations between the inputs and the output variable which probably might be visualized when plotting the training points of one predictor against the output, or it might be linear (the underlying truth is linear) with some noise added to the input. We will try different machine learning methods including linear and tree-based models and pick the one with the lowest out-of-sample error.

(4) Feature set optimization:
Sine the number of input features are limited a best subset approach will be applied to find the best set of predictive features. Also, strong evidence of interaction between input variables are present, hence interaction terms will be added to the feature space in order to enhance the prediction accuracy. A validation set will be used for these processes.

(5) Sensitivity to the training size:
In such practical problem domains, it might be consuming and more expensive to measure labeled data, and it is much easier to acquire unlabeled data. The size of the training set will be varied to study the behavior of the models and their need to the labeled and at which point they still can generalize well using limited amount of training data. We will also study the effectiveness of semi-supervised approaches and whether we can use unlabeled data to enhance performance.

(6) Sensitivity to the noise:
A measurement noise by non-accurate sensors will be mostly present in such industrial environments. Thus, it might be critical to take into consideration the sensitivity of the chosen models to corrupted-noisy training data and whether they will still generalize relatively well.

2.2. Our Prior and Related Work ----- None

2.3. Overview of Our Approach

2.3.1 Main Topic: Comparing The performance of Multiple Models Using a Supervised Approach, and Comparing the Training Size and Noise Sensitivity.

The optimal set of features will be firstly found using best subset approach and utilizing the interaction behavior to extend the feature space for a better generalization, a linear regression model will be used in those procedures. The data with the extended feature space will then be used to learn different models including regularized and more complex piecewise tree based models, and find their optimal hyperparameters. The chosen models will be compared according to their performance on an independent drawn set. The two baseline models that will be used as references to compare our generalization performance are:

- Output mean: A trivial regressor that outputs the average of the output variable.
- KNN: a simple lazy regressor that outputs the average of the labels of the instances in the same neighborhood.

The size of training data will be varied in the training process from 10% to 100% of the original training size in order to compare the 'data hungriness' of the models, and which models will still perform well when trained with relatively low training size. The models will also be compared according to their robustness to noise by adding a gaussian noise with varying power to the training set and observe which model will still have a reasonable accuracy. These two extra tasks could be considered as extension tasks, but they will be joined with the main topic since they will be applied to every considered model in the training process.

2.3.2 Extension: Semi-supervised Approach

In the presence of limited labeled data and plethora of unlabeled data, we will observe how a co-training semi-supervised approach will significantly enhance the performance when the unlabeled data are also used in the training process. Specifically, our KNN baseline performance trained on the limited labeled data will be compared to the performance of two KNNs learnt using different views with the inclusion of unlabeled points.

3. Implementation

3.1. Data Set

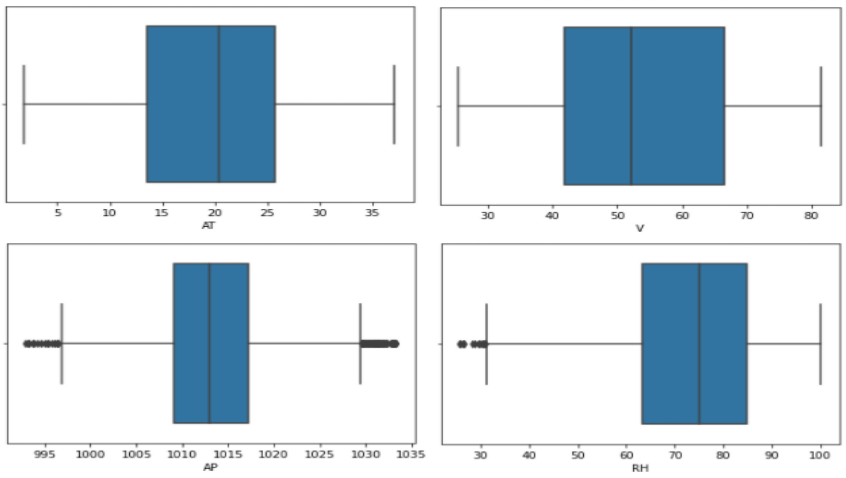
The dataset contains 4 numeric features in addition to one numeric output variable.[1]

Feature Name	Type	Description
Input Features		
Ambient Temperature (AT)	Numerical	The measured temperature in Celsius which ranges from 1.8 °C to 37 °C.
Exhaust Vacuum (V)	Numerical	The steam pressure measured in Centimeters of Mercury and ranges from 25.3 cm Hg to 81.5 cm Hg.
Ambient Pressure (AP)	Numerical	Atmospheric pressure of the turbine measured in Minibars with the range 992.9-1033 mbar.
Relative Humidity (RH)	Numerical	Relative humidity of the gas turbine as a percentage from 25% to 100%
Output		
Electrical Energy (EP)	Numerical	The full load electrical power output measured in mega watt ranging from 420.2 MW to 495.7 MW.

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

3.2.1. Outliers

After splitting the test dataset, the outlier points will be checked. Assume that a sample will be considered an outlier if one of its features values differ from the mean by a significant amount (e.g., 2.5 standard deviations). After applying that, the features 'AP' and 'RH' seem to have 51 and 7 respectively. A box plot from Seaborn package can also be used to visualize potential outlier points:

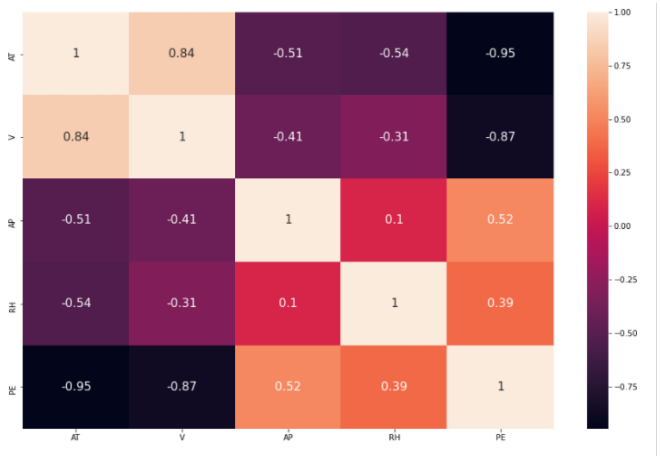


A suitable approach might be to compare the performance of the learnt models including and excluding those outliers to check their significance on generalization performance. However, since their number contribute a very small proportion of the data, they will be simply neglected.

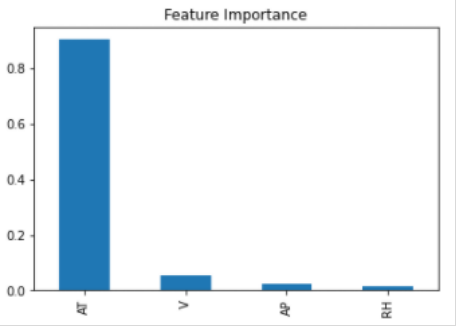
3.2.2. Dimensionality Adjustment

(a) Best Subset Feature Selection:

Since the number of features are limited, it might be useful to utilize a brute force approach that searches through the whole possible combination of features and find the best one with the most accurate prediction capability. A Correlation map using Seaborn package is firstly generated using the training set to give some indication of feature importance and correlation with the output.



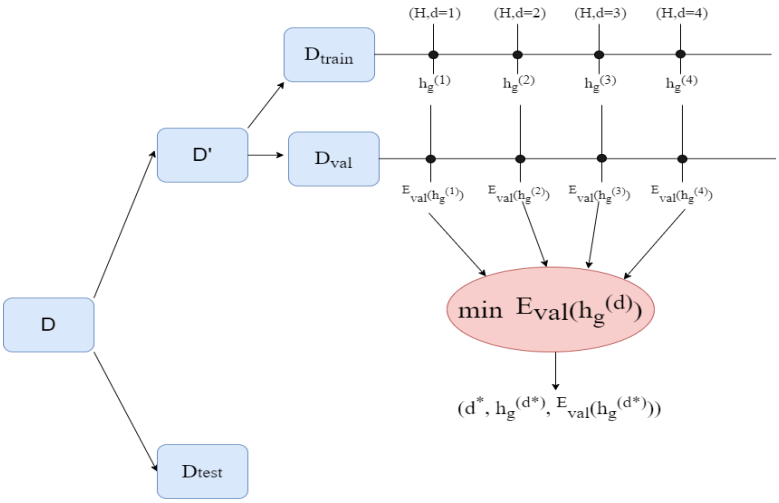
It seems that 'AT' and 'V' are more powerful predictors compared to 'AP' and 'RH', we will most probably expect that the best subset will be including these two features. Also, a decision tree regressor from was fit to the pre training data to quantify the feature importance and what feature was mostly used for splitting to predict the output variable:



Which lines up with the correlation matrix. We can probably conclude that 'AT' will be a very powerful feature for output prediction, and we might

check the relative performance of learned models including and excluding it.

The training data is used to fit a linear regression model with varying feature length. The validation set is then used to find the lowest error hypothesis with d features. After that, the minimum error among the lowest error hypotheses is found in addition to the corresponding features and feature length d. The following diagram describes the performed process, where (H,d=i) is the set of the linear regression hypotheses with i features, $h_g^{(i)}$ is the lowest error fitted model with i features using the training set, $E_{val}(h_g^{(i)})$ are the validation errors, and d^* is the optimal set of features.



The result of the validation errors is summarized in the table below:

AT	29.624962215873182
V	69.41876464874666
AP	206.12446998044672
RH	240.71969012375638
AT,V	23.929064025973837
AT,AP	29.109611878945188
AT,RH	22.737486500049457
V,AP	60.532197119062275
V,RH	64.35921951530476
AP,RH	175.93439363472552
AT,V,AP	23.303396907720217
AT,V,RH	20.25351265155221
AT,AP,RH	22.72544827067302
V,AP,RH	55.291041167491066
AT,V,AP,RH	20.159142369664234

We notice that the model with the full feature set was the optimal one. We also can see that the best performing single predictor model is the one with feature 'AT' as expected. The double feature model with 'AT' and 'V' is close to the full model in term of error performance, which lines up with our expectation as 'AT' and 'V' are more predictive and correlated to the output than the other two features.

The above results are repeated using a cross validation approach instead of using a validation set as it gives relatively more sense to the variability of an independent test set drawn from a distribution. The training and validation sets are combined to form the 5-fold cross validation set.

AT	29.870556396449114
V	70.84121166761975
AP	212.41100267583633
RH	247.26337895174584
AT,V	24.84356172142508
AT,AP	29.328474006255504
AT,RH	23.371352981071247
V,AP	62.024201081372794
V,RH	66.30001302148898
AP,RH	179.5226928299177
AT,V,AP	21.16888442980228
AT,V,RH	20.25351265155221
AT,AP,RH	23.361739826232395
V,AP,RH	57.084359628519884
AT,V,AP,RH	21.080411069019682

Similar to the validation set approach, the best performing model was the full model. The cross-validation approach gave little higher errors probably because the errors were calculated using a larger number of points, or probably one of the folds had a relatively higher error which increases the average of the error of the 5 folds. It might be generally expected in machine learning that the higher number of features might lead to overfitting issues, but in this data set, the full feature set model gave the best generalization and approximation to the underlying function.

(b) Interaction Terms:

As seen in the correlation matrix above, there exist high values of correlation coefficients between input variables (e.g. 0.84 between AT and V) which might give indication that there are some synergy between the features. These synergies will be modeled using interaction terms between every possible combination of two features. When a multivariate

regression model is learnt using the training data, we get the following result:

OLS Regression Results

Dep. Variable:	PE	R-squared:	0.928
Model:	OLS	Adj. R-squared:	0.928
Method:	Least Squares	F-statistic:	2.455e+04
Date:	Sun, 14 Nov 2021	Prob (F-statistic):	0.00
Time:	15:36:54	Log-Likelihood:	-22517.
No. Observations:	7654	AIC:	4.504e+04
Df Residuals:	7649	BIC:	4.508e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	452.8410	10.963	41.307	0.000	431.351	474.331
AT	-1.9731	0.017	-114.394	0.000	-2.007	-1.939
V	-0.2365	0.008	-28.917	0.000	-0.253	-0.220
AP	0.0639	0.011	6.005	0.000	0.043	0.085
RH	-0.1581	0.005	-33.684	0.000	-0.167	-0.149

Omnibus:	827.729	Durbin-Watson:	2.036
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4159.582
Skew:	-0.408	Prob(JB):	0.00
Kurtosis:	6.518	Cond. No.	2.13e+05

The low p-value of each variable implies its significance in predicting the output variable. Specifically, the probability of seeing t statistic $t = \frac{W}{standard\ error}$ extremer than what was observed is very small, so we reject the null hypothesis indicating that the input variable is insignificant w_i=0, and we conclude that there is evidence for the significance of feature in predicting the output value.

When we include all possible combination of interaction terms between the input variables ['AT*V','AT*AP','AT*RH','V*AP','V*RH','AP*RH'], where i*j indicates the interaction term between the feature i and feature j, and re-fit the regression model, we will get:

OLS Regression Results

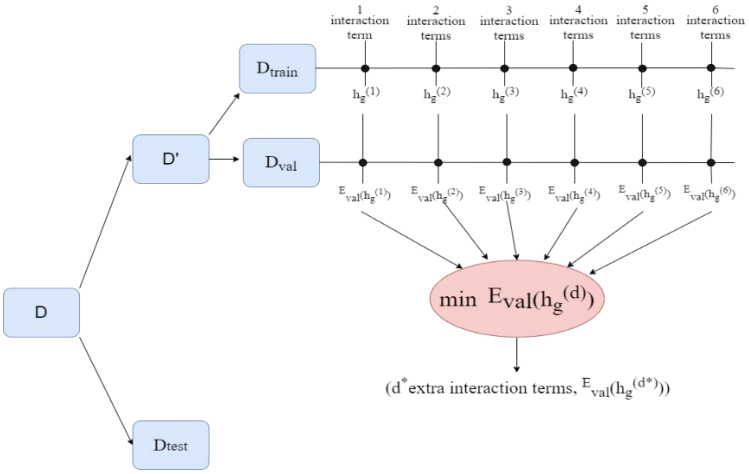
Dep. Variable:	PE	R-squared:	0.935
Model:	OLS	Adj. R-squared:	0.935
Method:	Least Squares	F-statistic:	1.104e+04
Date:	Sun, 14 Nov 2021	Prob (F-statistic):	0.00
Time:	15:36:44	Log-Likelihood:	-22095.
No. Observations:	7654	AIC:	4.421e+04
Df Residuals:	7643	BIC:	4.429e+04
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	679.3946	89.168	7.619	0.000	504.601	854.188
AT	-3.6471	2.698	-1.352	0.176	-8.935	1.641
V	-7.5210	1.521	-4.944	0.000	-10.503	-4.539
AP	-0.1481	0.087	-1.700	0.089	-0.319	0.023
RH	1.3993	0.881	1.589	0.112	-0.327	3.125
AT*V	0.0200	0.001	19.849	0.000	0.018	0.022
AT*AP	0.0011	0.003	0.430	0.667	-0.004	0.006
AT*RH	-0.0056	0.001	-6.121	0.000	-0.007	-0.004
V*AP	0.0067	0.001	4.476	0.000	0.004	0.010
V*RH	0.0007	0.001	1.333	0.183	-0.000	0.002
AP*RH	-0.0014	0.001	-1.654	0.098	-0.003	0.000

Omnibus:	1324.689	Durbin-Watson:	2.020
Prob(Omnibus):	0.000	Jarque-Bera (JB):	9499.451
Skew:	-0.638	Prob(JB):	0.00
Kurtosis:	8.306	Cond. No.	1.71e+08

We notice that the interaction model has a higher R-squared value compared to the basic model, and we can also infer that $\frac{0.935-0.928}{100-0.928} = 9.72\%$ of the variability in the output variable that remains after fitting the basic model has been explained by the interaction terms. It can be seen from the table that there are several interaction terms with low p-values which indicates their significance in predicting the output variable. It is worth noting that some of the original features have their p-values increased, but we will include all of the original features in feature set since it is difficult to interpret a model involving interaction terms without the original features.

A validation set will be used to find the optimal set of interaction terms. Since there are 6 possible pairwise interaction terms between input variables, a linear regression model will be used to find the best extra d interaction predictors such that d ranges from 1 to 6. The best performing models with 1 extra feature up to 6 extra features are then compared using a validation set. The above process and obtained results are summaries below:



	Best added interaction term	Validation Error
1 extra interaction term model	AT*V	18.90082030200786
2 extra interaction term model	AT*V, V*AP	18.70781361809116
3 extra interaction term model	AT*V, AT*RH, V*AP	17.81040156434748

4 extra interaction term model	AT*V, AT*RH, V*AP, AP*RH	18.12645020570941
5 extra interaction term model	AT*V, AT*RH, V*AP, V*RH, AP*RH	18.239206849245847
6 extra interaction term model	AT*V, AT*AP, AT*RH, V*AP, V*RH, AP*RH	17.983598677368415

Hence, a model with the four original features and the three extra interaction terms AT*V, AT*RH, V*AP was the best performing one. The following 7 features will be then used for the learning process:

AT	V	AP	RH	AT*V	AT*RH	V*AP

This new extended feature space will be used for all models that we want to compare since we found out that all models performed better when using this extended space compared to using the original feature space. For example, optimal parameter Ada boost gave a validation error of 10.56753 for the extended feature space as will be seen in training, while it gave 11.8372 using the original feature space.

3.2.3. Standardization

The mean and standard deviation in training set is calculated (to standardize each feature in training data with 0 mean and unit variance. Then, the validation and test sets are transformed using those standardization parameters. The standardized data will be used for the regularized models to compare the relative performance in error hyper-parameters selection process, and as a pre-process before adding the gaussian noise to check the models' robustness.

3.3. Dataset Methodology

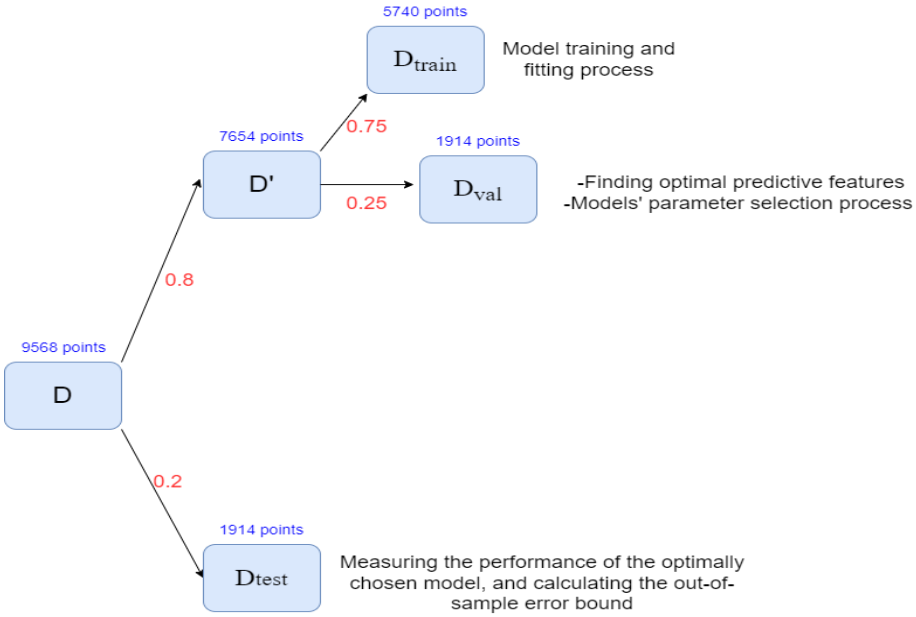
The dataset contains 9568 sample points. Both validation set and k-fold cross-validation approaches will be used throughout to compare the result of model selection and training process for multiple algorithms.

3.3.1 Validation set

The validation set was used for:

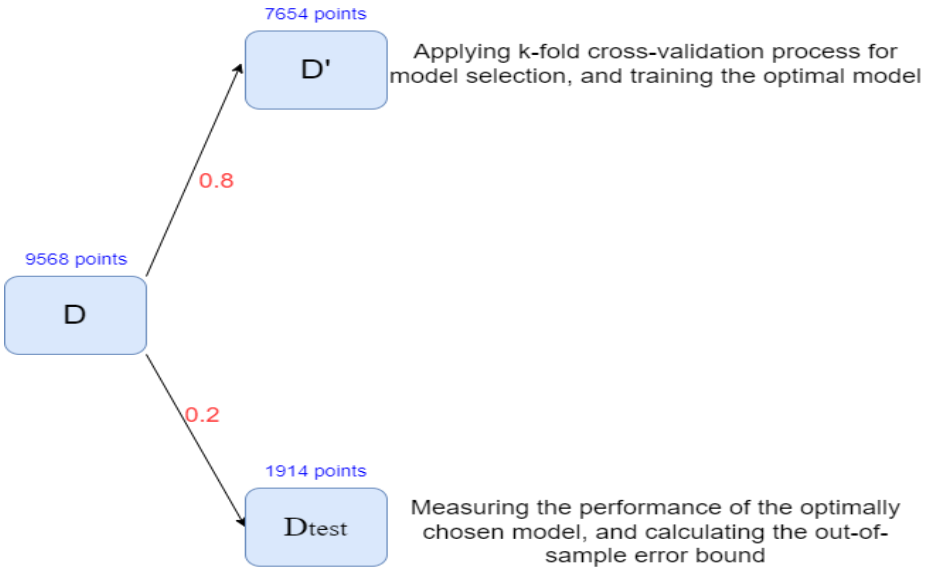
(a) Preprocessing and Feature selection: as seen in the previous sections, the validation set was used to find the optimal performing set of features and interaction terms

(b) Parameter and model selections: The validation set will be also used for hyper-parameter section for regularized regressions, random forest and other learning algorithms.



3.3.2 Cross Validation

A variant way for parameter selection process would be a k-fold cross-validation which might give a more accurate results since the model is trained with a larger number of points and the variance of estimated error is also reduced as it is less sensitive to a lucky draw of independent samples from the underlying population.



3.3.3 Test data

An independent set of 1914 sample points will be used as a test set. After obtaining the final model from different regression methods, the test set will be used to see the performance of the obtained model.

3.4. Training Process

For each algorithm, the following will be performed:

(a) Model Selection and Error Performance

The model will be trained, and the hyper-parameters will found through a model selection process using the validation set, and how sensitive the model to the parameter change.

(b) Training size sensitivity

In such industrial environment, it might be expensive and exhausting to acquire large amount of data. Thus, we will study the effect of varying the training size on the optimal chosen parameter and on the generalization performance. Specifically, Will the model still perform well even when trained with less data?

It is worth noting that the original feature space will be used here since we can't use the extra interaction terms because they were chosen in the pre-processing procedures using the original full training points.

(c) Noise Sensitivity

In data collection process of this dataset, multiple sensors were used to measure our variables such as the temperature and pressure. In such an environment, it might be typical that some noise are added by some of the sensors and the performance of the prediction models might hence be negatively affected. Thus, the sensitivity of the models will be studied by adding a gaussian noise with zero mean and varying standard deviation to the training set and observe the variability in the parameter selection and error performance. The noise is only added to the training data to reflect a situation where the training data was collected from a noisy sensor, and the test data was collected from another reliable sensor. The process of adding the noise is done multiple times and the error average was taken to account for the variability of gaussian distribution random-sample drawing. The original feature space will be used here due to the similar argument mentioned in (b).

3.4.1. Lasso Regression

It might be reasonable to start with simple linear regularized models such as Lasso and Ridge (in the next section) since we want to observe whether a linear model is sufficient to achieve a good performance, and to see whether the regularization can prevent overfitting when the model is lightly trained.

Lasso Regression adds a penalty to the objective function which is the absolute sum of the features' coefficients, and that prevents wild behavior of estimated model and will most likely lead to a better generalization compared to the unregularized case.

$$W^{*}=\underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \alpha \|w\|$$

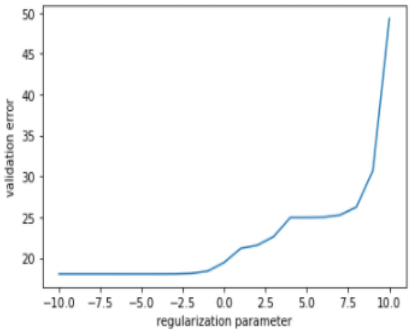
The optimal coefficients are found by optimizing this objective function. It is worth noting that the sharp edge of the constraint of the l_1 norm might causes feature selection since so there is a relatively higher probability that the constrained optimization will be satisfied in a point where some coefficients are zero which leads to a sparse solution in some cases.

(a) Model Selection and Error Performance

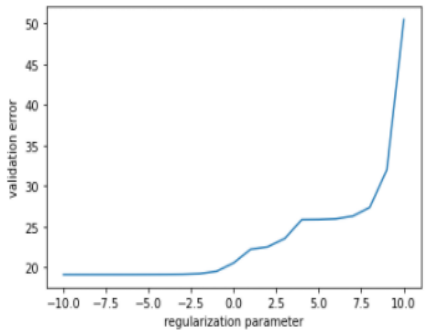
Unstandardized data

The regularization parameter α will be chosen within the range from 10 to 10 on the log scale, higher values correspond to higher applied penalty on the weights.

Validation



Cross-validation



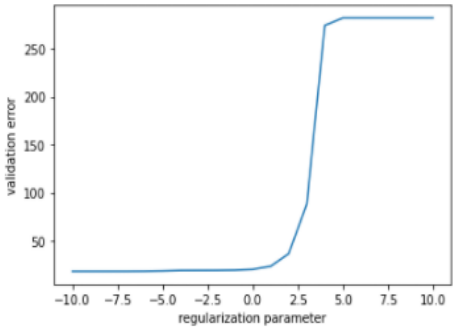
The best performing model for validation set approach was the one with α=0.03125 with validation error of 18.04232, while 5-fold cross validation gave α= 0.0009765 and mean error of 19.127 .

We notice that the amount of regularization in both approaches is very low and hence we got an error that is very close to the unregularized

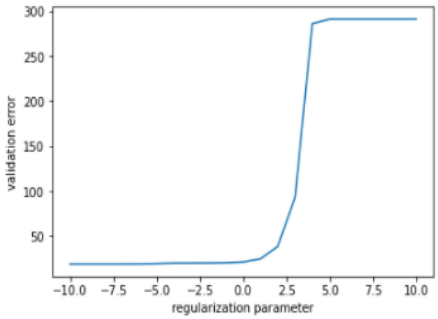
regression. This is probably because the unregularized model was not originally overfit due to the large amount of training and hence the model selection process tends to prefer the low regularization parameters. The cross-validation approach gave a higher error and a lower regularization because the model was trained and evaluated on more data which reduces the regularization parameter due to similar mentioned argument.

Standardized data

Validation



Cross-validation

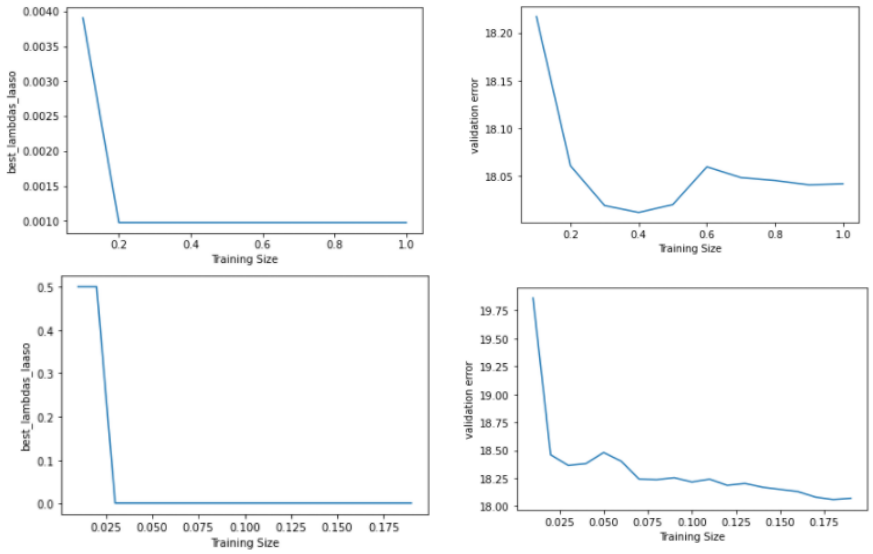


The best performing model for validation set approach was the one with α=0.001953 with validation error of 18.042433, while 5-fold cross validation gave α= 0.0009765 and mean error of 19.126032

The model performed almost equivalently when trained on the standardized or unstandardized data in terms of error performance on the optimal parameter. We notice that when the regularization parameter exceeds a certain value (approx. 2.5), the error heavily increases which is probably because when we apply higher penalties, the flexibility of the model won't be enough to approximate the complex underlying function.

For the standardized data, the error jumps very high when the model is heavily regularized compared to the unstandardized case, which means that for a certain parameter value, the model underfits the standardized data while it fits the unstandardized data relatively well, which might be probably because the extended feature space was chosen using the unstandardized data, so the standardized data will perform case will perform worse when the model is inappropriately regularized.

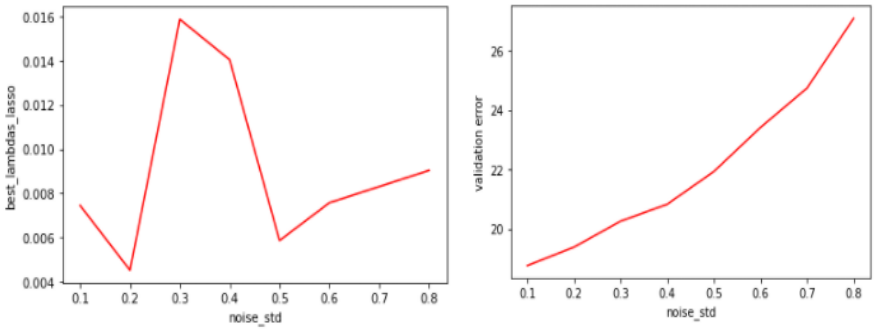
(b) Training size sensitivity:
Standardized/Unstandardized and Validation/Cross-validation gave almost similar results, hence only one of them will presented here:



The left plot is the selected parameter as a function of the training size percentage, and the right plot is the validation error against the training size. The second-row plots show the first row plots at very low values of training size to appreciate the details.

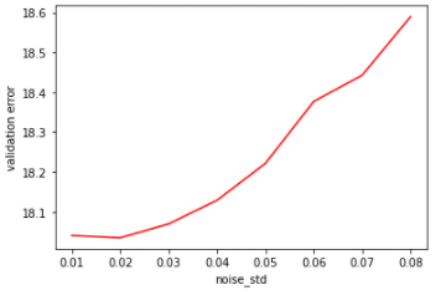
We notice that when the training size is low, the model prefers higher amounts of regularization in order to generalize well, which is expected since the low number of points will drive the model to be extremely wiggly and it hence needs large regularization to reduce the overfit and generalize better. When the training size exceeds approximately 0.1, there will be no regularization needed as there will be enough amount of data to prevent model overfitting. The validation error didn't vary much with the training size which means that the model was able to perform very well even with a very small training size.

(c) Noise sensitivity:



The left plot is average optimal parameter for different amounts of noise, the process of choosing the optimal parameter has been repeated 10 times to account for the variability of Gaussian noise independent sample drawing. We notice that there is small relative variation of the optimal as noise increase.

The right plot shows the validation error as function of noise. It can be seen that there is a steady growth of the error as noise increases, which gives some indication that Lasso will still perform relatively well on this dataset when the training data is affected by some noisy process. The model still performs well for low amount of noise as seen in the following figure because the regularized approximation model will not be highly altered when the training points are shifted by small amounts.



3.4.3. Ridge Regression

We will use another simple linear model to check its performance, and to compare its performance with Lasso, and to see whether the performance will differ using a different containing region for the optimal model's weights. Similar to Lasso regression, Ridge regression adds a penalty term to the loss function where the feature coefficients are constrained by the l2 norm region which might lead to a better out-of-sample performance since it prevents data overfitting.

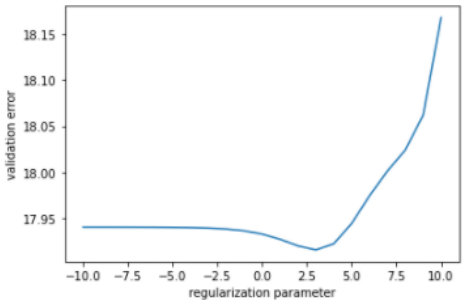
$$W^{*}=\underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \alpha \|w\|_2^2$$

(a) Model Selection and Error Performance

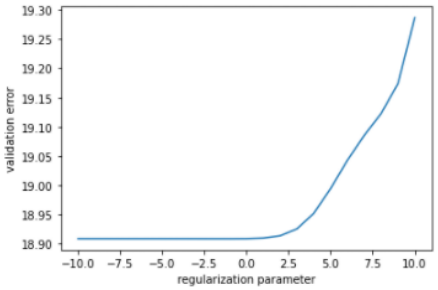
unstandardized data

Similar to Lasso, the regularization parameter α will be chosen within the range from 10 to 10 on the log scale, higher values correspond to higher applied penalty on the weights.

Validation



Cross-validation

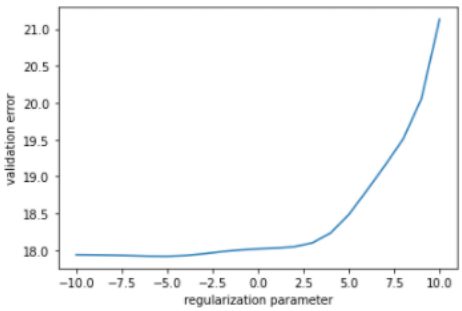


The best performing model for validation set approach was the one with $\alpha = 8$ with validation error of 17.915658, while 5-fold cross validation gave $\alpha = 0.5$ and mean error of 18.9079.

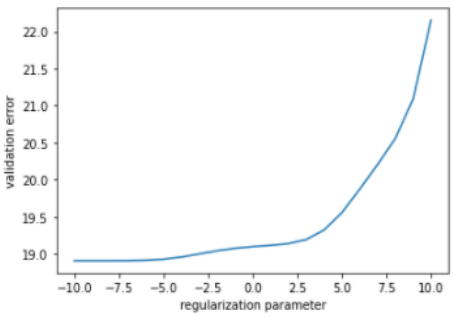
A similar patter of Lasso is noticed here where the error increases when the model is more regularized. In Ridge, however, the error begins increasing when we exceed approximately 2.5 on the log scale (compared to 0 in Lasso) which means that we are able to apply relatively more amount of regularization here and still perform well, which might give an indication that Ridge will generalize better than Lasso which is evident comparing their validation performance.

standardized data

Validation



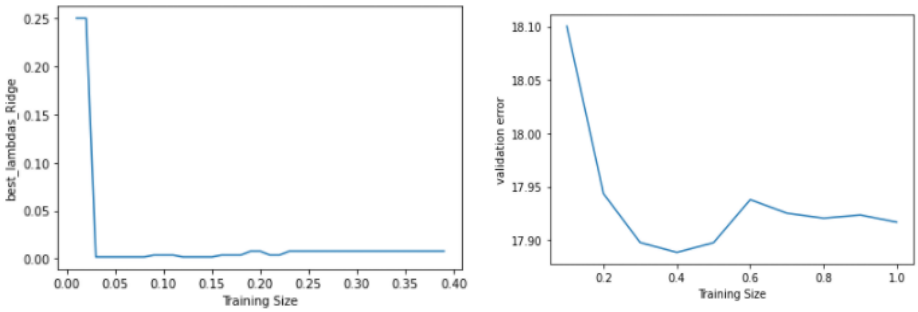
Cross-validation



The best performing model for validation set approach was the one with $\alpha = 0.03125$ with validation error of 17.91681, while 5-fold cross validation gave $\alpha = 0.001953$ and mean error of 18.9079.

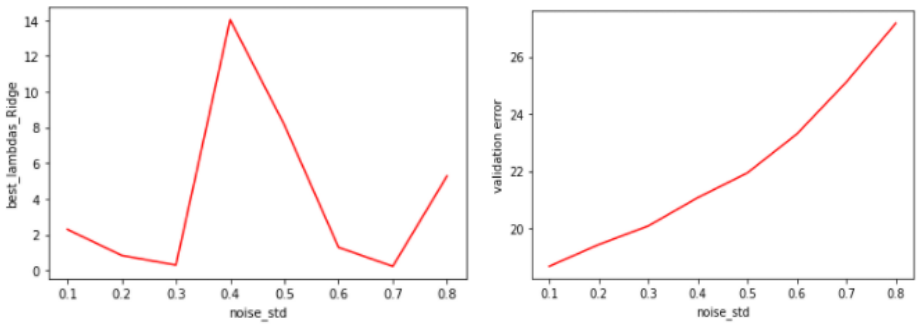
The error performance on the optimal regularization parameter is similar for the standardized data case. Lower parameter values are preferred here compared to the unstandardized case.

(b) Training size sensitivity:
We will present the standardized/ validation set approach as the other approaches gave approximately similar results.

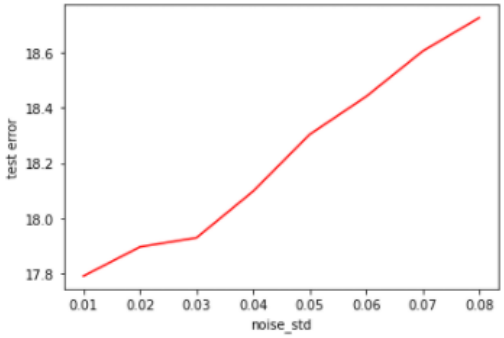


The left plot is the selected parameter as a function of the training size percentage, and we notice that in order for the model to generalize well, it is heavily penalized when the number of training points are few, which is expected since the complexity of the model was enough to overfit the data and hence a large regularization needed. The right plot is the validation error against the training size, and we can see that the model performed well even when trained with very few points which concludes that the model doesn't require many training points to perform well.

(c) Noise sensitivity:



The left plot is average optimal parameter for different amounts of noise, the process of choosing the optimal parameter has been repeated 10 times to account for the variability of Gaussian noise independent sample drawing. We notice that there is a higher variation of the optimal parameter in the noisy case compared to Lasso probably because the large amount of noisy data in Lasso led to sparsity effect without the need for extra regularization penalty.
The right plot shows the validation error as function of noise. Similar to Lasso, there is a steady growth of the error as noise increases, which means that Ridge will still perform relatively well on this dataset when the training data is affected by some noisy process. Similar to Lasso, Ridge also performs well for small noise amounts.

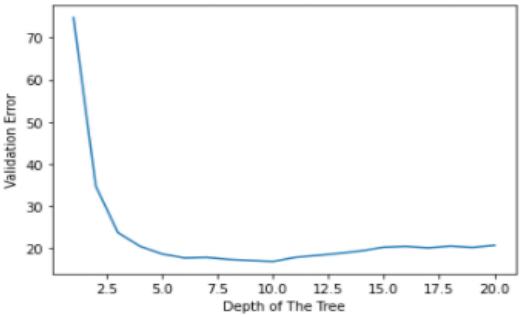


3.4.2. CART

CART is a symbol yet effective and interpretable regression model. It divides the feature space into regions by thresholding features at certain values. It is important to study the performance of this algorithms since it and its optimal performing parameter will be used in the coming more complex models.

(a) Model Selection and Error Performance

Trees with different depths will be learnt and the optimal performing depth will be found through a validation process.

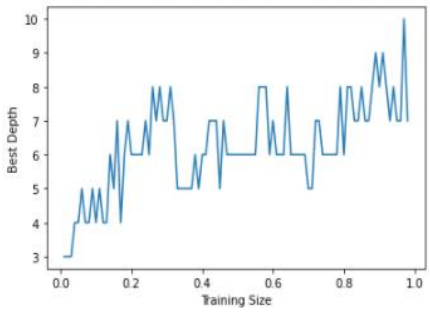


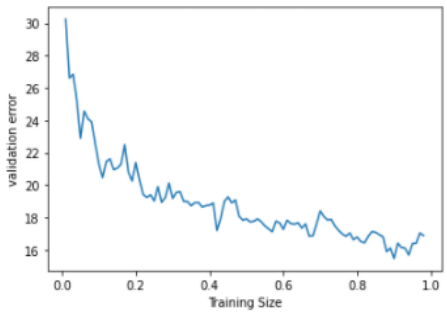
As seen in the error decreases with the trees' depth until it reaches 10 with a validation error of 16.9122, then it starts to increase which gives

indication that the model starts overfitting the data and generalizes less optimally. Thus, a depth of 10 will be used for the advanced regression and boosting methods.

(b) Training size sensitivity:

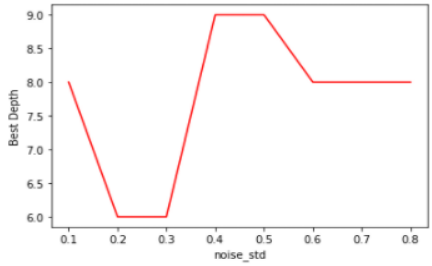
We will study the variability of the training size on the optimal parameter selection and on error performance

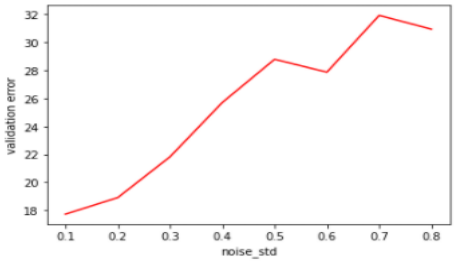




As seen in the plots, the validation error tends to decrease with the training size when it parameterized optimally. We can also see that the tree prefers a higher depth on average when trained heavily which means that as the training samples increase a higher complexity is needed in order to better approximate the underlying truth. We can infer that both error performance and parameter selection are highly sensitive to the training size compared to Lasso/Ridge since the regularization parameter in them prevents overfitting when trained lightly and make them still generalize relatively well.

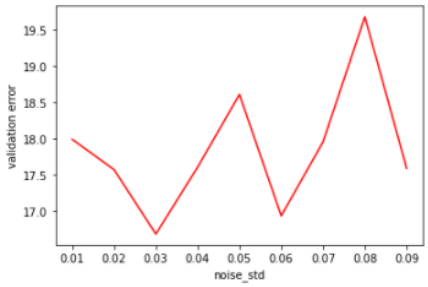
(c) Noise sensitivity:





We notice that the optimal depth is highly variable with the noise power and kind of depends on the drawing of the noisy gaussian samples. The validation error increases heavily with noise compared to the previous regularized regression methods which is probably because that CART will

continuously pick random noisy samples to split the feature space which will largely increase the prediction error. The error might rise significantly even for low noise values as seen in the following plot since a single tree is sensitive to outliers.



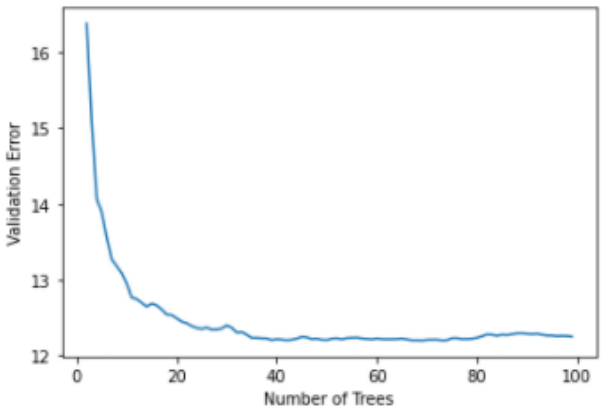
3.4.3. Random Forest Regression

Using the average prediction of multiple uncorrelated trees gives a better performance than using a single tree. Thus, we will try the random forest algorithm. It operates by constructing several decision trees during training time and outputting the mean of the results as the final prediction. We will use the optimal found depth for each tree, and the total number of trees will be taken as the hyper-parameter that needs to be optimized. The other parameter are:

- Best depth of the tree: 10, which was optimally found using a single CART through a validation set, higher values might make the forest prone to overfitting.
- Bootstrap = True, which is generally better to reduce the variance of the predicted output.
- Max number of features in each draw= None, since the number of feature to split in a tree is limited.
- Error Criterion = Minimum Squared error.

(a) Model Selection and Error Performance

A forest with varying number of trees will be learnt, and the optimal number will be found by a validation set measure. We expect that Random Forest will perform better than a single CART since it exhibits a lower variance which gives a better predictive power and accuracy that a single tree.



The optimal number of trees was 48 with a validation error of 10.767809 which is the lowest we got so far compared with the previous algorithms. It seems that from the graph that the error almost saturates when the number of trees reaches approximately 30 which means that 30 or more trees sufficiently split the feature space in a way that fairly approximate the true model.

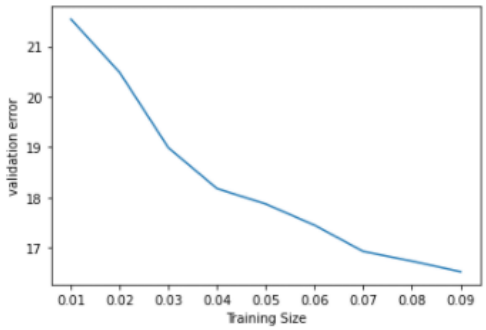
(b) Training size sensitivity:

For each training size, the optimal parameter and error will be found. The experiment will be repeated 10 times for each number of trees with a different draw of the bag size each time. The standard deviation of error will be also calculated in order to quantify the variability of the error with the different draws.

Percentage of the training size	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Optimal number of trees	40	46	41	43	47	43	50	48	41	48
Validation error	15.03944	13.940	13.073	12.5737	12.033	11.6218	11.4641	11.01578	10.9842	10.767809
Standard deviation of the error	0.5331	0.474	0.3304	0.30799	0.3464	0.2575	0.20245	0.19262	0.1177	0.0764

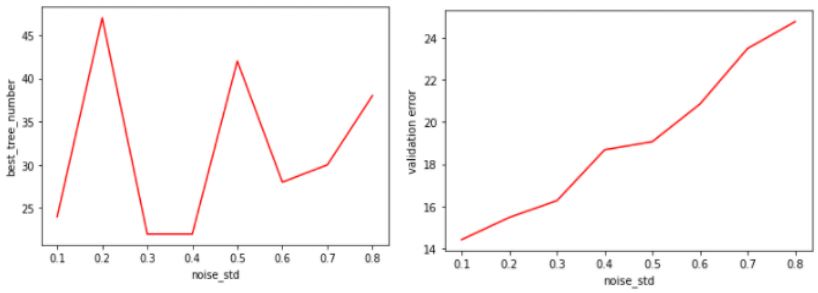
We notice that for all training sizes, higher number of trees (>40) are preferred to produce the minimum validation error, but the error seems to saturate when the number exceeds approximately 30. The error keeps decreasing with the training size, which is probably expected since adding

more training points can be seen as a regularization effect which generally leads to a better generalization. We also notice that the model performs reasonably well for small training sizes, even for extremely small sizes that are not listed in the table (e.g. 0.01 training size gave 22.247 error when enough number of trees are used, see the following figure) .

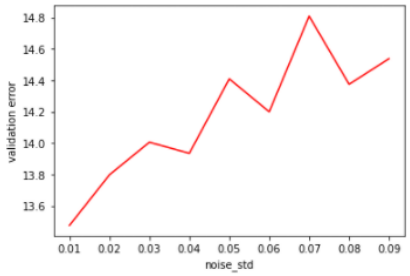


Intuitively, the standard error keeps decreasing with the training size since the variability of the drawing decreases as bagging size increases. It can be noticed that it is also low even for small training size which gives indication that the data is well distributed, and the error don't vary much with the random training draws. Comparing with a single tree case, the multi tree case will perform much better for lower training sizes.

(c) Noise sensitivity:



It seems that the number of trees varies randomly and heavily with the amount of noise and luckiness of the noisy samples draw, and as in the single CART case the error increases with the noise power. Random Forest, however, seems to have a relatively higher tolerance for small amount of noise (less than 0.3) compared to CART, then the slope of the error significantly increases, probably because the averaging operation of multiple CARTs which will give a better prediction that a single noisy tree. The system will still be able to perform well for lower amount of noise as seen in the following plot.

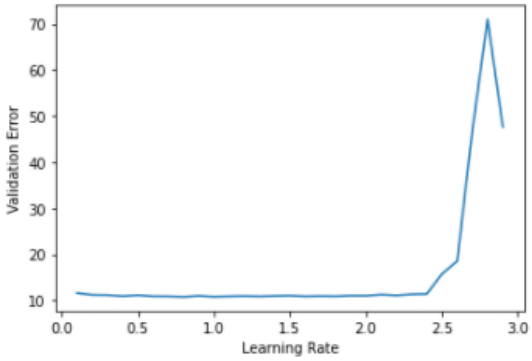


3.4.4. Ada Boost

Another ensemble tree-based, Ada boost, will be used since it has the advantage that it weights base regressor according to their performance on the weighted dataset, as if it discards bad performing regressors, which might give a better accuracy than the previous models. Ada boost is an ensemble method that creates a strong regressor from a number of weak regressors that perform better than random. A Single Cart with the optimal found parameters will be used as our base regressor in the Ada boost algorithm.

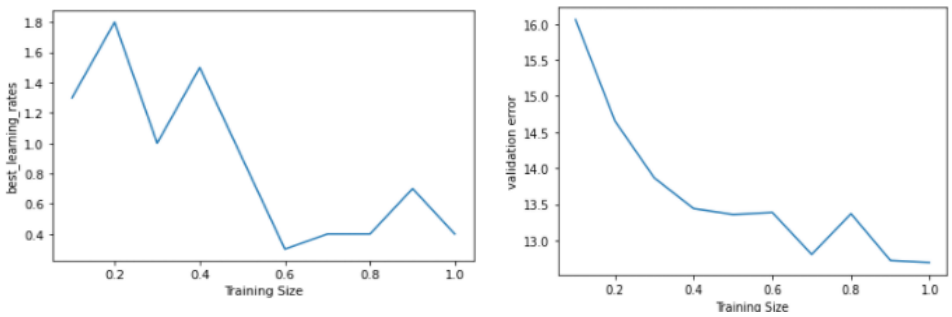
(a) Model Selection and Error Performance

A 48 base regressors with a maximum depth of 10 (optimal performing hyper parameters found from the precious models) will be used, and we hope that we achieve a better accuracy by finding the optimal learning rate (amount of the contribution with the new model to the existing one).

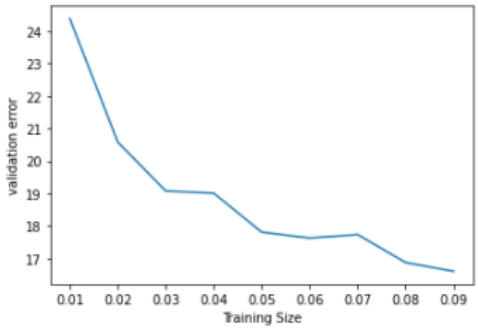


The lowest validation error was 10.56753 with a learning rate of 0.8. When the learning rate gets considerably large, the model overfits the training data and the validation error substantially increases. Ada boost has the best error performance although it is very close to Random Forest.

(b) Training size sensitivity:

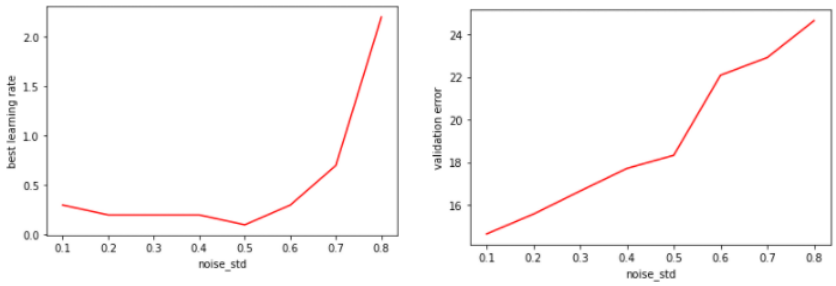


It seems that lower learning rates are preferred for higher training sizes which is probably because the base regressors are more weighted due the higher prediction error and the model is more easily overfitted for small training sizes. As expected, the error keeps decreasing with training size and, as in Random Forest, the model is able to generalize relatively well even when trained with small proportion of labeled data as seen in the following figure



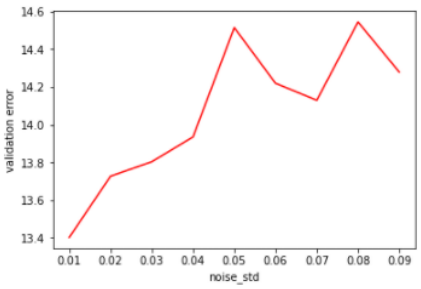
(c) Noise sensitivity:

For large amount of noise, the noise dominates the data the prediction error increases



The optimal learning rate increases for severely noisy data since the error increases. The model performs reasonably well until the noise reaches

approximately 0.5, then the error significantly increases, which gives indication that Ada boost is more noise tolerance than Random Forest probably because the weighting of the base regressors allows the bad performing regressors to be nearly ignored in the final prediction. For lower noise, Ada boost and Random Forest performed almost equivalently since the noise didn't affect the training strongly.



3.5. Model Selection and Comparison of Results

The final model will be chosen from the optimal model found from the training and model selection process taking into consideration:
(1) The error performance of the optimal model.
(2) Data hungriness which will be quantified by evaluating the error of the model when it is trained using only 10% percent of the training data.
(3) Noise robustness which will be quantified by evaluating the performance of the model when it is trained with the data mixed with 0.01 standard deviation gaussian noise, because all algorithms performed poorly for higher amounts for higher amounts of noise.
The validation error performance will be compared with the two baseline regressors:

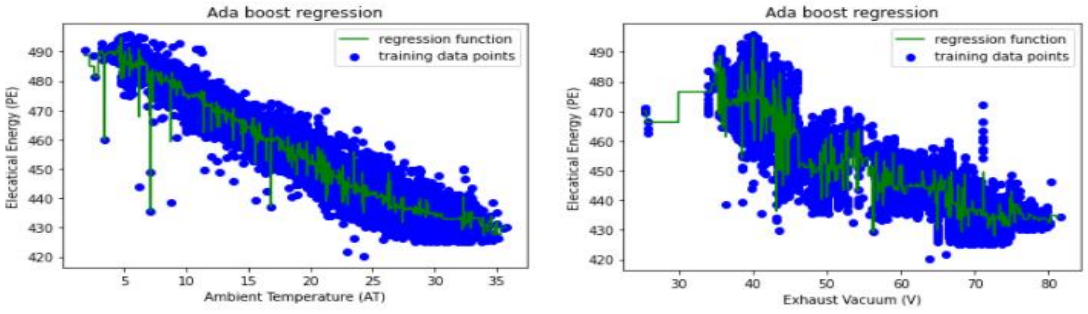
- Trivial regressor: always outputs the mean of the output variable which had an MSE of 281.971848250
- Simple lazy learning regressor: KNN has been used which has an MSE of 17.11741324

Methods	Model Selection	MSE Performance	MSE when trained using 10% of the training data	MSE when trained using noisy data
Lasso Regression	α=0.001953	18.04232	18.1962	18.0684
Ridge Regression	α= 0.03125	17.91681	18.1145	17.9875

CART	Tree's Depth=10	16.9122	22.3142	18.1486
Random Forest Regression	Number of trees=48	10.767809	15.03944	14.4617
Ada Boost	Learning Rate=0.8	10.56753	15.7479	13.6922

All models performed better than the trivial regressor which is expected since they used the covariates to learn. However, we notice that non-linear models (tree-based) performed better than linear models which is expected since there are a non-linear association between the input variables and the output, Ada boost was the best among them in terms of MSE. Linear models, however, requires much less training data and still generalize well and are less sensitive to noise because they have less complexity and the regularization parameter plays an important role to prevent overfitting the small sample sizes, while the complexity of decision regions of the non-linear models makes them more vulnerable to go through every training point and hence generalizes worse. It is worth noting that all models performed better using the extended feature space (with the interaction term included) compared to the original feature space. Ada boost performed better than Random Forest in the noisy case probably because the weighting of the base regressors allows discarding the bad performing ones in the final prediction, while Random Forest averages all outputs. Random Forest, however, requires less data to enhance the error performance because of the lower degree of freedom involved which makes it less prone to overfitting compared with Ada boost. We will pick Ada boost as our final model since we mostly care about the MSE criteria.

The 2D plots of the final regression function using the most powerful predictors (AT) and (V) are plotted below by plotting the predicted training points of the two features, each in a 1-D plot.



4. Final Results and Interpretation

The test set will be used to see the performance of our final model, which is Ada boost with the parameters:

- Maximum depth of the tree: 10
- Number of base regressors: 48
- Learning rate: 0.8

Methods	Test MSE
Chosen Ada boost	11.200788
Simple Baseline: KNN	16.904161
Trivial Baseline: output mean	292.512482

Intuitively, our chosen model performed better than the two baseline regressors since it learned from the features and had a higher complexity in defining the regression function. However, the test MSE of the simple baseline was not that far from our model, which is probably due to high correlation and prediction power of our features with the output variable and due to the low complexity of the underlying truth, which means that a simple interpretable regressor that utilizes those powerful features will be able to perform well. Higher complexity models, however, will have more accurate approximation to the regression function and hence generalizes better by a relatively good percentage. As mentioned before, linear models such as Lasso and Ridge generalize well, but the non-linear models (tree-based) performed better due to the existence of some non-linear association between the input features and the output variable which means a non-linear more complex model will generalize even better if it doesn't overfit, and this is why an ensemble method such as Ada boost performed better in terms of MSE. It is expected that when non-linear terms (e.g. quadratic, cubic) are added to feature space, we will have a lower MSE. For example, using deep learning approaches such as neural networks might develop optimal high dimensional feature space set that will perform better than the presented results. The time complexity of our model might be relatively high sense it uses multiple base regressors with non-trivial depths, but this might not be an issue due to limited number of training points and features. For the generalization to the unseen data, the test error using the R2 score will be used to calculate an upper bound for the out-of-sample error $E_{out}(hg)$ assuming the loss bound B_L is 1 since it is the highest possible error value (R2 score error), and our tolerance $\delta=0.1$:

$$E_{out}(hg) \leq E_{test}(hg) + B_L \sqrt{\frac{1}{2N} ln \frac{M}{\delta}}$$

$$\Rightarrow \quad E_{out}(hg) \leq 0.03983 + \sqrt{\frac{1}{2x1914} ln \frac{1}{0.1}}$$

$$\Rightarrow \quad E_{out}(hg) \leq 0.03983+0.0245257 = 0.0643557 \quad \text{with probability} \geq 1-\delta$$

Which will most probably be a less lossy bound than what the VC training bound will be. The generalization bound was low because of the high number of test data and the low hypothesis size (M=1)

5. Implementation for the Semi-supervised Approach

5.1. Data Set

Same data set as in the main topic.

5.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

Original feature space will be used here since we are not aiming to achieve an optimal error performance. Instead, we just want to compare the relative performance when using and discarding the unlabeled data.

5.3. Data set Methodology

A Similar data set division will also be used here, but only the training and validation sets are used here since we only want to compare the error performance of two approaches.

5.4. Training Process

In the presence of limited number of labeled data and many unlabeled data, two approaches will be compared:

(1) The error performance of our simple base regressor (KNN) trained using only the labeled data.

(2) The result of a semi-supervised approach where two regressors are trained on two views and they use the unlabeled data for training in a co-training manner. KNN will be used for the two views since it provides a confidence measure of prediction by consulting the influence of the labeling of unlabeled examples on the labeled ones [2].

We know that in order for the co-training approach to perform well, we need two conditions:

- Each view is sufficient to achieve good performance.
- The two views are conditionally independent.

As mentioned, the original feature space will be used here since it will be hard to divide the interaction terms in the two views because they are obviously correlated. Assume the first view are the features AT and AP, and the second view are the features V and RH. Since ‘At’ and ‘V’ are powerful predictors as seen in the table in 3.2.2, the two views will provide good prediction results.

mvlearn.semi_supervised package was used to do this process and the code was modified from:[3]
https://mvlearn.github.io/auto_examples/semi_supervised/plot_cotraining_regression.html

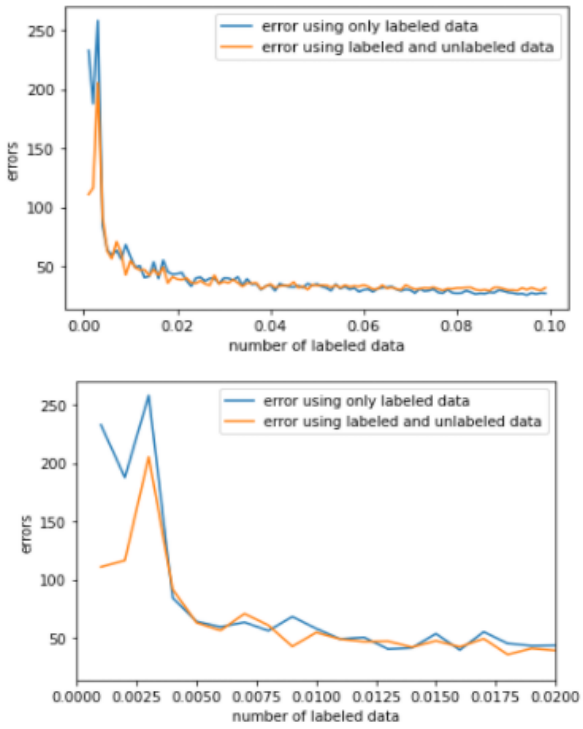
The model in the second view will use the most confident labeled points by the first view model in its training process, and vice versa until we label all the unlabeled data. Then, the resulting model will be tested on the validation set to see its error performance and compare it to the supervised case where only the labeled points were used for training.

5.5. Comparison of Results and Interpretation

For an extreme case where the number of labeled data is 6 points, we have that:

Methods	MSE
trivial regressor (mean of labeled data)	281.55986489028214
supervised KNN (trained with labeled data only)	168.4605636739816
semi supervised co-training (trained with labeled and unlabeled data)	107.81720521717348

We notice that utilizing the unlabeled data has significantly improved the performance. It is worth mentioning that as the number of labeled points increase, the supervised approach will begin to perform better than the semi-supervised approach which is probably because the co-training assumption is violated since it is obvious from the correlation map that the two views are not conditionally independent. Obviously, the above results vary with the draw of the labeled training set, but semi-supervised approach performed better on all trails for very small labeled training points. Also, we inferred from the previous sections that the data are highly descriptive and the regressors are able to generalize well even when trained with few points. The following chart shows the performance of the two approaches:



The second plot shows the first plot at finer low values.
When the number of labeled points is low (approximately below 287 labeled points), the semi-supervised approach has a lower error compared to the supervised approach. When the number of labeled points gets larger than that, they will sufficient themselves to accurately approximate the underling function and have a slightly better performance than the semi-supervised case. Probably the violation of the co-training assumption that the two views are independent causes it to perform slightly worse.
One can argue that the semi-supervised approach didn't show much significant in this problem domain since we are able to achieve good performance even with very limited labeled training size, due to the low complexity of model and high predictive performance of the features.

6. Summary and conclusions

Here are some summarized main findings from the analysis in previous sections:

- Ambient Temperature (AT) was the most important factor in determining the electrical energy output (PE). Specifically, the models involving (AT) performed much better in the validation error sense.

- The pre-processing showed the existence of significant synergy between features and a linear regression model was used to find the optimal interaction terms between features, and those extra features improved the performance for all models.
- There exists a high correlation between the input and the output variables, and the high predictive power of some of the features allowed the lower complexity models to achieve a good performance.
- The linear models required less data to still perform well since the regularization terms prevent overfitting the small training set. Non-linear models, on the other hand, are more prone to overfitting due the high complexity and multiple parameters involved, and hence they require relatively more training to still generalize well. This also applies to the noise analysis due to the same argument.
- Ada boost was the best performing model and it is not that sensitive to parameter change as it will still perform reasonably well when the optimal parameters are altered.
- The validation set approach gave approximately similar results to the cross-validation approach in terms of MSE and optimizing the optimal parameters which might give indication that there is no high variability in the learnt models since the data in the underlying population is consistence.
- The semi-supervised co-training approach was extremely useful in enhancing the performance when the percentage of labeled data is small. When the number of labeled data is higher, a supervised approach will be efficient to produce a good performance since we inferred that our models don't require much training to perform well.
- We noticed that we didn't experience a lot of overfitting signs since the training size was sufficient and that plays important role to prevent our models to overfit the data, i.e. the sample complexity is higher than the model complexity (our models' complexity was not enough to fit every single point due the high sample size)
- The semi-supervised approach performs little worse than the supervised approach probably because the violation of the SSL assumption (two views conditional independence) and due to the good result of the supervised case even when lightly trained.
- We saw that adding extra features to our feature space has enhanced the generalization performance, and we expect that if we add extra non-linear features (e.g. quadratic) will additionally improve the accuracy. A future approach might be to use a neural network to develop a set of optimal features, where we will probably have even better accuracy at the cost of lower human interpretability.

7. References

[1] Archive.ics.uci.edu. 2021. *UCI Machine Learning Repository: Combined Cycle Power Plant Data Set*. [online] Available at: <https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant>

[2] Zhou, Z. and Li, M., 2007. Ijcai.org. Available at: https://www.ijcai.org/Proceedings/05/Papers/0689.pdf

[3] Mvlearn.github.io. *2-View Semi-Supervised Regression — mvlearn alpha documentation*. Available at: <https://mvlearn.github.io/auto_examples/semi_supervised/plot_cotraining_regression.html>

8. Appendix

The extracted zip folder contains those files, where:

- main: The code that loads the chosen model with its optimal parameter and outputs the test set result.
- Final model: the final model parameters that will be loaded by the main file.
- Main Topic: The code for the main topic: Comparing The performance of Multiple Models Using a Supervised Approach, and Comparing the Training Size and Noise Sensitivity.
- extension topic: the code for the extension topic: Semi-supervised Approach.
- Dataset: the dataset used.
- X_test: the features values for the test set that will be loaded by the main file.
- y_test: the labels for the test set that will loaded by the main file.

```
├── main. ipynb
├── Main Topic.ipynb
├── extension topic.ipynb
├── final model.pkl
├── Dataset.csv
├── X_test.csv
├── y_test.csv
```